U.S. Army Research Institute
for the Behavioral and Social Sciences

Research Report 1533

AI Tools for Foreign Language Training

Merryanna L. Swartz and Joseph Psotka
U.S. Army Research Institute

July 1989

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency Under the Jurisdiction of the Deputy Chief of Staff for Personnel

EDGAR M. JOHNSON
Technical Director

JON W. BLADES
COL, IN
Commanding

Technical review by

Phil Gillis
George W. Lawton

NOTICES

~~**DISTRIBUTION:** Primary distribution of this report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, ATTN: PERI-POX, 5001 Eisenhower Ave., Alexandria, Virginia 22333-5600.~~

FINAL DISPOSITION: This report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS --
2a. SECURITY CLASSIFICATION AUTHORITY -- 2b. DECLASSIFICATION/DOWNGRADING SCHEDULE --	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ARI Research Report 1533	5. MONITORING ORGANIZATION REPORT NUMBER(S) --
6a. NAME OF PERFORMING ORGANIZATION U.S. Army Research Institute 6b. OFFICE SYMBOL (If applicable) PERI-IC	7a. NAME OF MONITORING ORGANIZATION --
6c. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600	7b. ADDRESS (City, State, and ZIP Code) --
8a. NAME OF FUNDING/SPONSORING ORGANIZATION U.S. Army Research Institute for the Behavioral and Social Sciences 8b. OFFICE SYMBOL (If applicable) PERI-IC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER --
8c. ADDRESS (City, State, and ZIP Code) 5001 Eisenhower Avenue Alexandria, VA 22333-5600	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO.: 62788A PROJECT NO.: 790 TASK NO.: 336 WORK UNIT ACCESSION NO.: H1

11. TITLE (Include Security Classification)
AI Tools for Foreign Language Training

12. PERSONAL AUTHOR(S)
Swartz, Merryanna L., and Psotka, Joseph (ARI)

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM 88/02 TO 88/11	1989, July	29

16. SUPPLEMENTARY NOTATION
--

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Artificial intelligence, Foreign language (SDW)
			Natural language,
			Intelligent tutoring systems

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Army needs to provide retention training in foreign languages for several military occupational specialties (MOS). With limited time and trainers to do so, developing computer training systems to support the training requirement is a viable solution. However, these computer systems need advanced technology such as Artificial Intelligence and natural language processing methods in order to deliver successful language instruction. We review the state-of-the-art technology in this report. Keywords. (U)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Joseph Psotka	22b. TELEPHONE (Include Area Code) (202) 274-5540	22c. OFFICE SYMBOL PERI-IC

DD Form 1473, JUN 86 Previous editions are obsolete.

Research Report 1533

AI Tools for Foreign Language Training

Merryanna L. Swartz and Joseph Psotka
U.S. Army Research Institute

Technologies for Skill Acquisition and Retention Technical Area
Zita M. Simutis, Chief

Training Research Laboratory
Jack H. Hiller, Director

U.S. Army Research Institute for the Behavioral and Social Sciences
5001 Eisenhower Avenue, Alexandria, Virginia 22333-5600

Office, Deputy Chief of Staff for Personnel
Department of the Army

July 1989

Army Project Number
2Q162788A790

Training and Simulation

Approved for public release; distribution is unlimited.

FOREWORD

This report on artificial intelligence technologies for language training reviews the current methods and tools available for creating natural language training systems on computers. Military managers need to be aware of advanced technologies and their application to training problems. This review describes the technology base from which U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) scientists are working to develop foreign language training systems.

EDGAR M. JOHNSON
Technical Director

Accession For	
NTIS GRA&I	☑
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

EXECUTIVE SUMMARY

Requirement:

The U.S. Army Research Institute for the Behavioral and Social Sciences (ARI) conducts research on applications of advanced technology in training systems for the Army. One application involves developing natural language processing capability on computer systems for language training and machine translation. This report presents an overview of natural language processing technology applicable to these capabilities.

Procedure:

A literature review and evaluation of selected computer languages, tools, and systems was conducted.

Findings:

Results of the literature review and system evaluations are described in the report.

Utilization of Findings:

The purpose of this report is two-fold. First, the report describes various computer tools necessary for developing a computer training system for the Army to support retention skills in foreign languages. Second, the report provides military personnel with a general overview of natural language processing and artificial intelligence technology as applied to Army training systems.

CONTENTS

AI Tools for Foreign Language Training

Introduction

Several Army military occupational specialties (MOS) have a foreign language requirement. Training newly assigned soldiers in relevant languages imposes a continuing problem of maintaining productivity in the schoolhouse and maintaining language skills on the job. Soldiers arriving from a training school often have good general language skills but little background in a specific job assignment in the foreign language. Subject matter knowledge, such as preparation of the battlefield, operational orders, and interrogation are unusually not covered in the initial language instruction the soldiers receive. For the newly assigned soldier, there is an imposing array of difficult terminology and new linguistic structures to acquire. As a result, additional language acquisition as well as language skill retention are required of the soldier, coupled with learning new MOS related skills. However, hard-copy training materials to provide the soldier with the additional language training are difficult to develop and maintain across the 26 languages the Army teaches. Nevertheless, several compelling technologies are reaching a state of ready application to solve these training problems.

Current advances in artificial intelligence and natural language processing technology provide a unique opportunity for the development of intelligent computer-assisted language learning (CALL) to support foreign language instruction in the Army. CALL has been used in foreign language learning for several years with varying degrees of success (Smith, 1987). However, computer technology has recently advanced to a point that makes the presentation of natural, contextualized language use feasible in a computerized delivery system.

The specific focus of our research requirement at the U.S. Army Research Institute is the development and application of these high technology advances to the teaching of foreign languages in the Army schoolhouse. The principle technologies we view as relevant to this training problem are natural language processing (NLP) and intelligent tutoring systems (ITS). Tools and formalisms that support this technology to structure knowledge as it is presented and acquired in an intelligent CALL include hypertext, grammar formalisms, parsers, and algorithms and models for presenting and teaching skills in intelligent, adaptive environments. Criticisms to CALL (Smith, 1987) state that only certain of the four basic language skills (reading, writing, speaking, hearing) are supported in this environment. While this argument is valid, we propose that for those skills supported in CALL systems, an instructional foreign language environment built with natural language processing representations and using hypertext to structure the knowledge can improve language skill acquisition and retention in these areas. Such a CALL system, further embellished by intelligent tutoring system technology and even audio-visual support, with video disc or CD-ROM, is envisioned as a support to traditional foreign language training in the classroom.

The foreign language tutoring system we envision will provide students with multimedia interactions that support listening, reading, and writing skills. Students will interact with the system by typing in responses to realistic computer "dialogues." The computer will be equiped to understand these

responses and provide sound adaptive instruction as needed by individual students. Sound for language samples will be provided on command to retain students' listening skills. Students could be encouraged to speak and repeat what is heard, although the computer would not be able to process spoken input. Nevertheless, intelligent CALL promises to provide Army foreign language teachers with a multimedia language lab environment suitable for exercising the student in various drill and practice activities that are prerequisite to active language production skills.

In this report, we review the tools, technologies, and components for developing intelligent CALL. We close with a discussion of an intelligent CALL prototype that exploits these technological advances. An appendix listing a glossary of technical terms and their definitions is provided to assist readers with the content of this report.

Hypertext Tools for Structuring Instructional Knowledge

Hypertext software provides a text-based system for computer applications that goes beyond simple text to include graphics, video, and sound as well as links between cards of information and cross references that result in networks or graphs of information structures. Hypertext systems provide the user with a variety of paths through the information structure. Command menus are usually used to help traversal through some hypertext space. A very real question worthy of further research is how to best organize the knowledge for presentation using hypertext software and tools. This is especially critical when hypertext is used to develop instructional systems (Russell, Moran, and Jordon, 1988).

The structure of semantic memory has been analyzed by psychologists in models of networks of concepts linked together in a coherent, associated structure (Anderson, 1983; Collins & Quillian, 1969). This work is ideally suited for cognitive modeling in a hypertext computer system. Within such a network model, the structure of concepts and the links between them are dependent on the relationship of features and default values stored at each concept node. The question of the psychological reality of these descriptions has been subordinated for the time being to the need to explore the utility of these representation schemes for instructional purposes. However, work on techniques for describing cognitive structures using hypertext models is proceeding steadily within the domain of cognitive science.

For example, in the application of hypertext as a tool for creating computer-based foreign language instruction, we can use hypertext models to hypothesize about the organization of vocabulary in memory (Swartz, in press). Similarly, hypertext can be used to develop actual instructional modules. Hypertext, coupled with more powerful natural language technologies on the computer, promises to provide the means for creating new, advanced technology training systems.

Natural Language Processing in Computer Systems

Communicating with computers via natural language interfaces remains a very difficult research problem. This is because understanding free-form language input by computer systems involves considerable work in developing databases that have information about the context, word knowledge, and linguistics. Nevertheless, current work in natural language processing and linguistic analysis has developed representation formalisms and parsing strategies that

may enable some progress in solving the natural language problem. In order to develop such an interface and further extend it in a computer environment that can understand and respond to not only English, but any other natural language, we must first understand the natural language itself and how it is used in communication. Thus we must study human conversation, discourse processes, and language comprehension and production skills. While this paper will not review the psycholinguistic parameters pertaining to foreign language learning, we mention it as one of the essential components in developing intelligent CALL. After all, how can we represent natural language in a computer system without an understanding of language use in situ?

Natural language understanding computer systems have been developed (Winograd, 1972,) and much recent work has occurred in this area to exploit the earlier advances (Allen, 1987). A computer system that can not only understand but generate language must be capable of several essential functions. The system must perform 1) lexical analysis (vocabulary understanding), 2) morphological analysis (word formation), 3) syntactic analysis (grammar understanding), 4) semantic analysis (contextual meaning), and pragmatic analysis (language use in context). These analytic processes depend not only on language understanding, but on real world knowledge and context in which to ground any potential language utterance. In a computer system, this requires that specific kinds of knowledge necessary for these analysis procedures be adequately represented. Indeed, we have described previously how knowledge representation with hypertext systems for language instruction is of critical importance. However, the domain knowledge to be taught, the language or "grammar", must also be represented in some formalism so that the computer can access this knowledge during an instructional sequence. Thus we discuss a variety of grammar representations which address this issue of formalisms in natural language processing applications. These strategies will be briefy explained below.

The representation of linguistic knowledge alone without contextual information and a process to understand the language will not solve the natural language problem for intelligent CALL, however. We also need a mechanism for matching the language input a student provides with the grammar representation in the computer system. This process is called "parsing" and we will see that different kinds of parsing can be performed depending on the context surrounding the language input and system-dependent efficiency methods for the particular parsing strategy used.

Technological advances in this subdiscipline have provided powerful formalisms for representing natural languages in computer systems. These representations promise to address the issues of lexical, morphological, semantic, and syntactic analysis required by an intelligent CALL environment. Parsing strategies which access and manipulate the language representations will address syntactic, semantic, and pragmatic (to some degree) analysis. An important question is which combination of grammar formalisms and parsing strategies will provide a suitable learning environment in intelligent CALL? By finding the best combination, we can develop a tutoring system that provides quick, correct analysis of student input in the foreign language. The decision on what combination of grammar and parser to use is based on several criteria. These criteria include providing a system that is psychologically realistic. For example, the computer should be able to parse "colorless green ideas sleep. " and inform the student that while grammatically correct, the

sentence is meaningless. Secondly, instructional goals for the system will be used to decide how comprehensive a grammar is required or how detailed a parse is required (i.e., is key-word matching enough or must the system understand complex grammar structures?). Thus we review several current grammar formalisms and parsing strategies below to provide an overview of these technologies.

Grammar Representations

Semantic Grammars. A semantic grammar is based on a simplified conceptual dependency construct (Burton and Brown, 1979). The grammar is organized as a set of attribute-specific semantic categories for a given domain. This construct has been used in early natural language applications for handling question asking about some specific topic. In NLP the grammar defines the allowable patterns that can be matched with a given input string. These legal patterns restrict the form of the natural language which cuts down on ambiguity and provides surprisingly good language coverage. A sample semantic grammar and a query to parse might look like the following:

<s> ——> What is <ship-property> of <ship>?

<ship-property> ——> the <ship-prop> <ship|prop>

<ship-prop> ——> speed length type
<ship> ——> <ship-name>

<ship-name> ——> Kennedy|Kitty|Hawk

Query: What is the length of the Kitty Hawk?

A semantic grammar is context-free and can be parsed by any of the several existing context-free parsers (Earley, 1970). Context-free means that word order is not taken into account. The contextual semantics or meaning of a given, ordered input string are derived from the word categories in the grammar. Key-word and pattern-matching strategies are used to parse and answer queries. While this mechanism allows for natural language interaction, there is no flexibility for syntactic deviations in question form. Lexical variations will also be constrained to the vocabulary represented in the semantic grammar.

In order to get more flexibility in natural language input and parsing strategies than is provided in a semantic grammar, separate representations will need to be developed to represent the lexicon, (word dictionary), morphological tables (word formation), phrase structure rules (rules used to derive sentences from input strings), and grammar. This brings our discussion to another set of natural language processing representations. While the unification formalisms to be discussed next allow for rich natural language interaction with computer systems, they pose a different kind of problem in parsing in regard to semantic and pragmatic language contexts.

Unification Formalisms. Grammar formalisms are representations used to describe a language, its set of sentences, the syntactic features of the sentences, and their semantics. These formalisms have been constructed to help linguists understand universal linguistic theory as well as provide a means for

A LEXICAL-FUNCTIONAL GRAMMAR REPRESENTATION

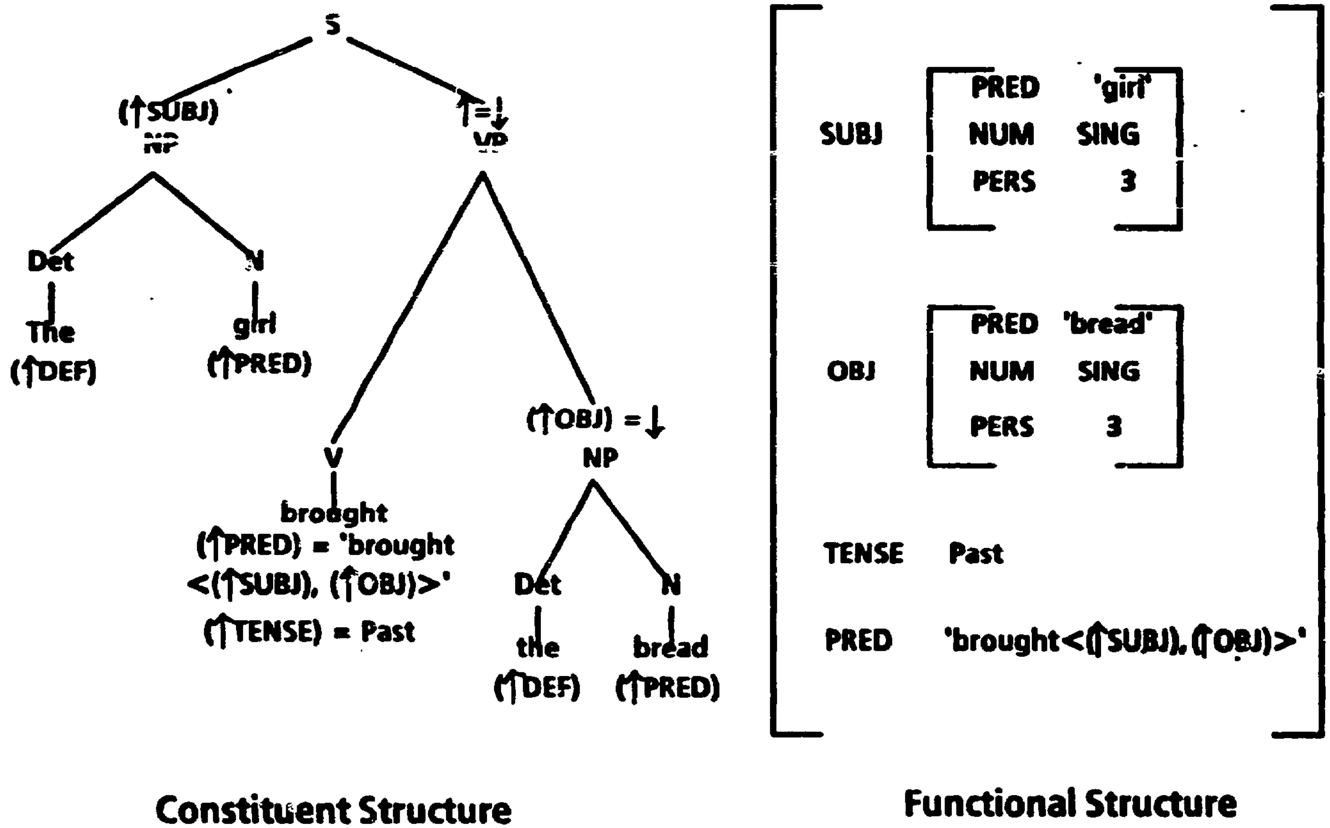


Figure 1. The constituent structure illustrates the word tree for the elements in the phrase with indicators pointing to their functional counterparts. The functional structure shows the functions (subject, object, predicate) for the constituents and their attribute values (number, person).

computational interpretations of the grammar being studied when it is represented in some computer system. A unification formalism is one that prescribes a given operation (pattern- matching) between an input string (a sentence) and the grammar. Schreiber (1986) discusses the essential assumptions of a unification formalism:

1) it is surface-based with a direct relation to the surface order of the sentential elements

2) it associates the sentence string elements with some relevant information (knowledge) domain

3) it is inductive, defining the association between an element and the information domain recursively and according to a specified inference procedure

4) it is declarative, defining permissible associations, not how the association is computed

5) it is context-feature-based in that associations between features and values are taken from a well-defined, structured set

Examples of unification-based grammar formalisms include: Functional Unification Grammar (FUG), Lexical-Functional Grammar (LFG), General Phrase Structure Grammar (GPSG), and Definite Clause Grammar (DCG). Each are briefly defined below. For more information the reader is referred to the references cited.

The FUG representation uses functional structures as generalized feature/value pairs (feature: verb, value: to see)to denote linguistic information in some language input. Patterns and constituent sets are used in the matching or unifying of the sentence elements with the grammar. FUG also introduces disjunction (allowing seperate rule analysis) into the grammar which allows for getting around any potential constraints imposed by simultaneous satisfaction of all applicable rules. In contrast to the LFG formalism below, FUG deliberately blurs the distinction between constituent and functional structures (Dowty, Karttunen, and Zwickey, 1985). This formalism emphasizes the functional description of the language with the constituent structure specified by a means of patterns for noun and verbal elements rather than phrase structure rules.

LFG was developed primarily as a linguistic theory (Bresnan, 1982) and then became recognized as a useful formalism for representing natural languages in computer systems. LFG is based on a model of syntax that is not completely structurally-based (Sells, 1985). The formalism describes grammatical functions (subject, object) which are represented by functional (F-) structures in the grammar. An F-structure for a given sentence describes the appropriate grammatical functions or role for the sentence constituents ("John" might function as a subject in one sentence and an object in another). Constituent (C-) structures (noun phrase, verb phrase) are the part of the formalism that represents the syntax of a sentence (see Figure 1). The lexical component of the theory emphasizes the commitment of LFG to characterize semantic relations between constituent elements. LFG can thus analyze grammaticality at the

lexical level. This formalism, more than the others described here, has been used to investigate psycholinguistic issues in relation to language processing systems (Ford, Bresnan, Kaplan, 1981).

GPSG also came into being from a linguistic analysis motivation. This formalism attempts to retain a formally restrictive system of surface feature structures that can handle a wide variety of semantic and syntactic phenomena. Unification procedures used to analyze language input are also highly restructured. A given phrase structure tree for a sentence will satisfy various rules in the GPSG grammar based upon specific principles (control agreement, head feature) specified in the formalism.

DCG arose from work in the PROLOG programming language (the other formalisms mentioned above are generally written in LISP). This formalism uses term structures rather than feature structures for a given sentence. Terms are the informational elements in DCG with the unification principles derived from the theorem-proving mechanism inherent in PROLOG. Examples of terms are:

s (head (subjHead, form)))

np (subjhead)

np (head (subjHead, form)))

Terms differ from the feature structures in FUG, LFG and GPSG in that they identify sentence constituent values by their order in the term structure rather than their association with the feature.

<u>Parsing Strategies</u>

Parsing is "searching a space of possibilities" in an input string (Kay, 1985) as the string is matched with the grammar. The discussion of parsing on computer systems will be facilitated by some simple definitions taken from Earley, (1970). A <u>language</u> is a set of strings made up of a sequence of symbols. These are referred to as <u>terminal</u> symbols that represent the words defined in the language. A <u>context-free grammar</u> is a formal device that specifies the strings in a given set. This grammar also uses <u>non-terminal</u> symbols that represent syntactic classes of words. A <u>sentence</u> is a string of several terminal and non-terminal symbols. When parsing occurs, an algorithm to search the string and make matches between its elements and the grammar is performed. There are many strategies one can use, transition network parsing, top-down parsing, bottom-up parsing, and deterministic parsing. Which approach is used depends on the domain and the goal of the natural "language" application.
Here we are interested in the needs of an instructional system for foreign language that requires quick, accurate parsing.

An augmented transition network (ATN) (Woods, 1970) is one way to parse language going from an initial state (the first word in a string) to a final state (the last word in a string). ATN's use the nodes in a network of language elements to represent parts of speech or phrase names. The arcs between nodes are used to implement certain actions when they are transversed (See Figure 2).

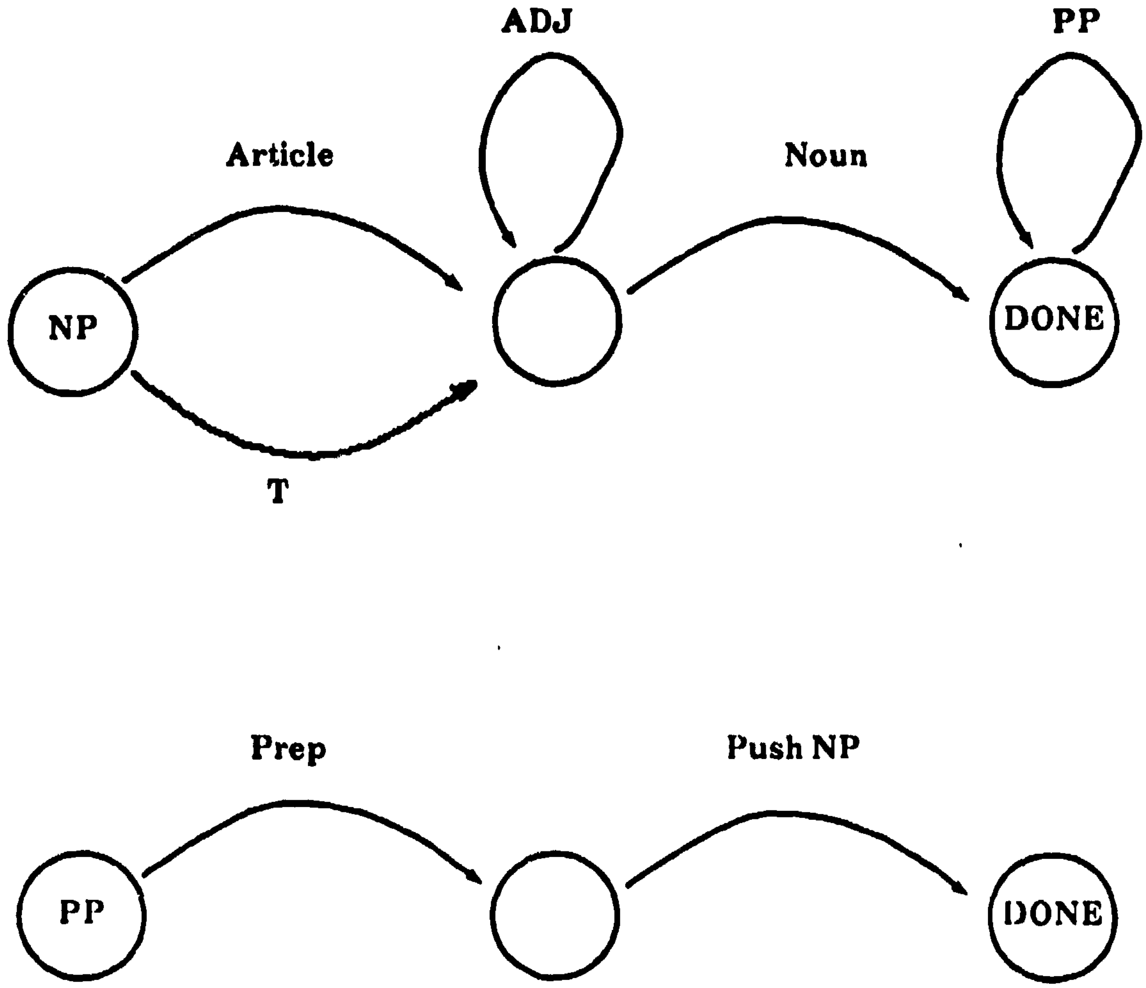


Figure 2. This ATN illustrates the states and transition arcs for parsing a noun phrase with a prepositional phrase. The circles represent states. The noun phrase has an article, possible adjectives (the ADJ loop arc on the middle state), and a noun with a loop for possible prepositional phrases. The second part of the ATN shows the states and transition arcs for parsing the prespostional phrase.

Augmenting such a network allows for three kinds of transitions:

1) Adding arbitrary tests to the arcs such as agreement of a word and its modifier.

2) Adding structure-building actions to arcs to be used by the parser to determine semantic analysis or active-passive sentence transformations.

3) Allowing subroutines to be called on arcs. For example, in Figure 2, a preposition (PP) is attached to a noun phrase by a subroutine call to PP.

While the ATN formalism provides many advantages for its use in natural language (conciseness of the representation and conceptual effectiveness) and can deal with complex linguistic phenomena), it has a high cost for processing speed. This is because ATN's are interpreted processes. Interpreted processes occur in real time and may be frustating for students since more time to parse ATN'S is needed.

Top-down parsing is based on the phrase structure rules it uses and is expectation or goal-driven. Given a string, the parser starts by applying a particular phrase structure rule, the goal, to elements in the input string. Conversely, bottom-up parsing is data-driven. This procedure starts with words in the input string and looks for phrase structure rules that will allow a parse. Figure 3 illustrates these two approaches schematically.

Marcus (1980) developed a deterministic parser (where only one next step is allowed for a given input) to avoid the backtracking problem required when sentence inputs involved complex clause construction and potential "garden path" or ambiguous parses. This means that the computer would have to keep track of the terminals in a string and "backtrack" to these symbols in case the sentence being parsed might have more than one meaning. Other new parsing strategies being developed include the use of the special features of logic programming in PROLOG to execute parses from a DCG representation. PROLOG handles backtracking problems very easily.

With this brief overview of grammar formalisms and parsing strategies, we can see that the technology exists for handling natural language on computer systems. In order to understand how this technology can be used to develop intelligent CALL for foreign language training, however, we turn to a discussion of the technology available for creating intelligent tutoring systems.

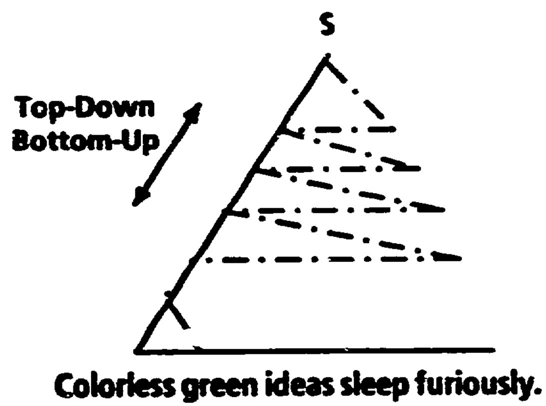
Intelligent Tutoring Systems (ITS) for CALL

Over the past five years, ARI has accumulated a great deal of practical knowledge about expert systems and intelligent computer-assisted instruction. This work has carried out studies and development efforts that have created working ITS in the domains of electronic troubleshooting, computer programming, and technical training (Simutis and Psotka, 1987; Psotka, Massey and Mutter, 1988). From these efforts, it is safe to say that several of the technology base domains are well developed and ready to apply in broad Army mission areas (Fletcher and Psotka, 1986). As ITS becomes more practical and effective,

PARSING

Parsing a sentence is searching a space of possibilities

Need a formalism to describe the structure of natural languages



Grammar

(SETQ GRAMMAR '((S NP VP)
(NP ADJ N)
(VI V ADV)
(VP VI)
(ADJ COLORLESS)
(ADJ GREEN)
(N IDEAS)
(VI SLEEP)
(ADV FURIOUSLY)))

Figure 3. In a top-down parser, the algorithm starts with a sentence (S) node and matches the constituents with the elements in the input string. This corresponds with the first rule in the grammar shown on the right, (S NP VP). A bottom-up strategy starts with the elements of the input string and matches them with their phrase structure rule counterparts. This corresponds to the grammar elements, (ADJ Colorless), (ADJ green), etc. moving upword to the (S NP VP) rule.

their applications are moving from more well understood and tractable domains (electronics troubleshooting, math) (Sleeman and Brown, 1982) to fuzzier complex skill areas such as foreign language learning. However, in addition to having the computational resources to develop these systems, we need to understand what goes on in good tutoring from the standpoint of the tutor, the knowledge taught, and the student.

A good human tutor is an expert in the domain being taught and an expert pedagogue. Being a good pedagogue implies a number of sophisticated skills. A good tutor is able to accurately monitor each trainee's knowledge state. That is, the tutor is able to determine what each trainee knows, does not know, and knows incorrectly. The second part of that skill, knowing where trainees typically go wrong, has turned out to be the key to developing good ITS. The tutor's knowledge of the trainee's level of knowledge extends in time allowing the tutor to make predictions about instructional level and pacing. The tutor has an extensive set of knowledge resources: books, graphics, films, carefully prepared courses of instruction, and the skill to use these appropriately. Finally, the tutor is able to marshall all these resources and skills in a carefully coordinated interaction with the trainee. These aspects of a good human tutor are analogous to the modules that make up an ITS: the expert model, the pedagogical model, the student model, and the interface. The first three will be discussed here in relation to developing intelligent CALL.

The Expert Model

The expert model in an ITS represents the domain-specific knowledge and the inferencing knowledge or reasoning processes involved in solving problems in the instructional domain (Clancey, 1986). As part of the domain-specific knowledge, a rich knowledge base of facts and skills the student will need to learn is represented. Often these representations take the form of semantic networks. The inferencing knowledge is a representation of the procedures an expert in the domain would use to reason about or apply to the domain-specific knowledge when solving a problem. This kind of procedural knowledge often takes the form of production rules.

In foreign language learning, the form of the domain-specific knowledge in the expert model, a particular foreign language, would be represented by one of the grammatical formalisms (FUG, DCG, LFG, GPSG) described above. The selected formalism would represent the grammatical rules and lexicon items with requisite features and properties that describe the semantics and syntax of a given language. This domain-specific knowledge is what a native speaker of the language knows. It would change from language to language. However, world knowledge that is needed to understand the contexts of the language in use must also be represented and tied in with the expert grammar. This knowledge might be represented in a semantic network or script-like representation to provide the context or situation in which the language to be taught is used. The inferencing knowledge in the expert model for a foreign language tutor would be a particular parsing strategy that would be applied to the grammar and lexicon to both generate and understand language within a given context. These parsing strategies would most probably take the form of rules and be a separate knowledge structure within the expert model.

The Student Model

The student model component in ITS is the information that describes a student's knowledge about what is being taught and that allows the tutor to adapt subsequent instruction to the student's needs (Clancey, 1986). Self (1974) refers to a student model as a set of programs designed to represent a student's knowledge state. VanLehn (1988) specifies that two components make up a student model: the structure of representation of the knowledge in the model, and the process that manipulates or transforms the structure. He further defines these two components as the model itself (structure), and diagnosis (process) as the mechanism for transforming the student model. Knowledge state assessment is therefore the outcome of this process. How the student model is formulated depends on measuring performance, identifying errors and misconceptions, and comparing this knowledge to that represented in the expert model of the target domain. All of these parameters are contingent upon how the domain knowledge to be taught is represented in the expert model of the tutoring system. Since foreign language learning is the domain of interest, the student model structure would be some subset of knowledge as represented in the grammar and lexicon in the expert model. Because the student model is derived from the interaction between the student, the expert model, and the pedagogical model in the tutoring system, the form of the representation of these individual models as well as the diagnostic process knowledge used for modeling a student are critical to the success of an ITS in the modeling task. Several methods for enabling the diagnostic process exist such as differential, overlay, and model tracing techniques. Clancey (1986) provides a good overview of these methods in existing systems.

The Pedagogical Model

This model represents the pedagogical strategies and lessons to be presented to the student. In addition, the diagnosis process plays a role in this model so that the computer tutor can present remedial lessons based on the diagnoses of student errors. This component of the ITS generates instruction based upon the student's instructional history during a session with the computer. The pedagogical model uses information from the diagnostic process to identify a student's particular instructional need and then adapts the instruction accordingly. This activity uses sophisticated planning techniques (Russell, 1988) to change pedagogical strategies and tactics according to a student's performance in a lesson. This adaptive approach differs drastically from more traditional computer-based training (CBT) which presents students with prestored, frame-based sequences of instruction.

The pedagogical strategies and tactics that must be represented in a foreign language tutor should represent methods appropriate for teaching in CALL environments and that are sensitive to the particular learning strategies involved in language learning. More importantly, this instructional knowledge must follow a principled, theoretically grounded approach to the language learning process. Research on how to rationalize such an approach for designing language instruction is currently underway (Swartz and Russell, 1989).

Prototype Intelligent CALL

We have described the existing technology and tools that are available for creating intelligent CALL. But how are these complex formalisms and tutoring modules to be integrated into a single coherent instructional system? The following is an overview of one such prototype.

A fair amount of basic research has been deployed with the goal of using NLP techniques for language instruction. Barchan, Woodmansee, and Yazdani (1986) provide a short review of the topic. An early attempt to use NLP techniques for language instruction (Feuerman et al., 1987) has shown great promise. The system uses a Spanish grammar written in the LFG formalism to drive an interactive dialogue-based tutoring system in beginning Spanish. The choice of LFG is fairly important although many other systems, such as FUG and GPSG now have very similar characteristics. The functional parsing approach used by LFG has a clear advantage over many other formalisms in that it gets all its semantic and syntactic meaning from the representation of the lexicon. This factor saves computational effort for parsing input strings when compared to other formalisms. Thus the interface the student uses provides timely responses to input typed into the system.

Feuerman's, et al, Computer-Assisted Language Learning Envinronment (CALLE) capitalizes on the use of hypertext to present the student with various instructional interactions. CALLE engages trainees through a number of functional hypertext windows shown in Figure 4. The main dialogue window provides the key functionality: on-line exercises in which trainees can engage in realistic foreign language dialogues with the machine. These dialogues are monitored by special rules to determine which immediate lesson goals have been exercised, and which remain to be tested. The system deliberately generates dialogue contexts to exercise the remaining instructional goals as the "conversation" proceeds. Other windows allow trainees to use the foreign language without being directed by the machine's specific dialogue goals (the "Try It Window"), but with diagnostic comments. Additional tools exist for the trainer to create scripts to drive the dialogue and new rules for presenting additional aspects of the language. Although CALLE does not have a sophisticated pedagogical model by ITS standards, we view it as a precursor for creating a complete ITS with robust tutoring strategies and adaptive presentation. While student errors are identified, CALLE lacks true student modeling capability. Nevertheless, the basic elements of a robust tutoring the system are present and can be upgraded.

CONCLUSION

The technologies reviewed here indicate that the time is ripe for research and development of advanced technology in foreign language training systems for the Army. ARI is currently actively engaged in such work.

CALLE was developed as a prototype tutor of linguistic knowledge using a combination of natural language processing, hypertext, and ITS technology to demonstrate the feasability of such an approach for language training. It was designed to strengthen the procedural skills of using a language in natural discourse and to encourage the self-correction of a trainee's grammatical rules and understanding in an interactive environment.

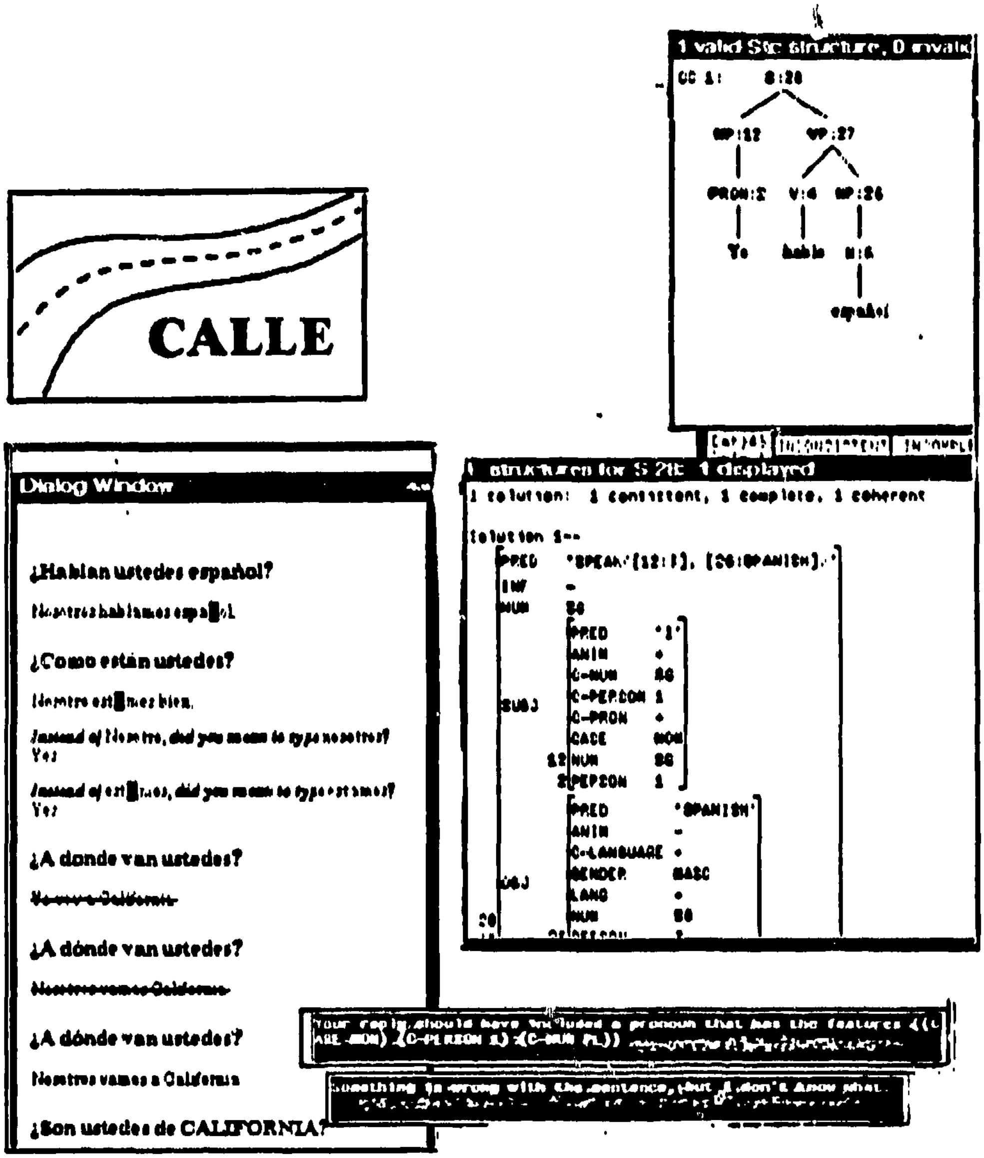


Figure 4. The windows shown here illustrate a typical dialogue exchange between the tutor and student. Errors are flagged and simple feedback provided. Other windows let the student see the parse tree for a phrase and even its functional structure.

While CALLE is not a complete foreign language ITS, it does demonstrate the potential for such a tutoring system for the Army schoolhouse language skill retention curriculum. Current research at ARI, both contractual and in-house, continues to investigate these technologies and systems for developing intelligent CALL. Not only will we explore the potential of these technologies for language learning, but basic cognitive and psycholinguistic research is planned to assess the usefulness of structuring knowledge for instructional presentation and its effect on the recall, retention, and use of language within robust, advanced technology training systems.

REFERENCES

Allen, J. (1987). Natural language understanding. Menlo Park, CA: The Benjamin/Cummings Publishing Company.

Anderson, J.R. (1983). The architecture of cognition. Cambridge, MA: Harvard University Press.

Barchan, J., Woodmansee, B.J., and Yazdani, M. (1985). A Prolog-based tool for French grammar analyzers. Instructional Science, 14, 22-66.

Bresnan, J. (Ed.) (1982). The mental representation of grammatical relations. Cambridge, MA: MIT Press.

Burton, R.R. and Brown, J.S. (1979). Toward a natural-language capability for computer-assisted instruction. In H. O'Neill, Procedures for instructional systems development. New York, NY: Academic Press.

Clancey, W.J. (1986). Qualitative student models. Annual Review of Computer Science, 1, 381-450.

Collins, A., Quillian, M. R. (1969). Retrieval time from semantic memory. Journal of Verbal Learning and Verbal Behavior, 8, 240-47.

Dowty, D.R., Karttunen, L., and Zwicky, A.M. (1985). Natural language parsing: Psychological, computational, and theoretical perspectives. Cambridge, UK: Cambridge University Press.

Earley, J. (1970). An efficient context-free parsing algorithm. Communications of the Association for Computing Machinery, 14, 453-60.

Feuerman, K., Marshall, C., Newman, D., and Rypa, M. (1987). The CALLE Project. CALICO Journal, March, 25-34.

Fletcher, D., and Psotka, J. (1986). Intelligent training systems for maintenance. Signal, 89-95.

Ford, M., Bresnan, J., and Kaplan, R. (1981). A competence based theory of syntactic closure. In: J. Bresnan (Ed.), The Mental representation of grammatical relations. Cambridge, MA: The MIT Press.

Kay, M (1985). Parsing in functional unification grammar. In: Dowty, D.R., Karttunen, L., and Zwicky, A.M. (Eds.). Natural language parsing: Psychological, computational, and theoretical perspectives. Cambridge, UK: Cambridge University Press.

Marcus, M.P. (1980). A theory of syntactic recognition for natural language. Cambridge, MA: The MIT Press.

Psotka, J., Massey, L.D., and Mutter, S.A. (Eds.) (1988), Intelligent Tutoring Systems: Lessons Learned. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Russell, D., Moran, T.P., and Jordan, D.S. (1988). The Instructional Design Environment (IDE). In J. Psotka, L.D. Massey, and S.A. Mutter (Eds.), Intelligent Tutoring Systems: Lessons Learned. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Russell, D. (1988). The IDE Interpreter. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Self, J.A. (1974). Student models in computer-aided instruction. International Journal of Man-Machine Studies, 6, 261-276.

Sells, P. (1985). Lectures on Contemporary Syntactic Theories. (Lecture Notes). Stanford, CA: CSLI.

Shieber, S. (1986). An Introduction to Unification-Based Approaches to Grammar. (Lecture Notes). Stanford, CA: CSLI.

Simutis, Z.M., and Psotka, J. (1987). What we know that we didn't know five years ago: An overview of army research in intelligent computer-assisted instruction. Proceedings of the National Science and Communications Center, Fort Gordon, GA.

Sleeman, D., and Brown, J.S. (1982). Introduction: Intelligent tutoring system. In D. Sleeman, and J.S. Brown (Eds.), Intelligent tutoring systems. New York: Academic Press.

Smith, W.F. (1987). Modern media in foreign language education: Theory and implementation. Lincolnwood, IL: National Textbook Company.

Swartz, M.L., and Russell, D.M. (1989). FL-IDE: Hypertext for structuring a conceptual design for computer-assisted language learning. Instructional Science, 18, 1-21.

Swartz, M.L. (in press). LEXNET-INSITU: Development of a hypermedia foreign language vocabulary learning invironment. ARI Technical Report, Alexandria, VA: The Army Research Institute.

VanLehn, K. (1988). Student modeling in Intelligent Tutoring Systems. In R. Lawler and Yazdani, M. Intelligent tutoring systems: An overview. (1986), Expert Systems.

Winograd, T. (1972). Understanding natural language. New York: Academic Press.

Woods, W. (1970). Transition network grammars for natural language analysis. Communications of the ACM, 13, October.

APPENDIX A

Glossary of Technical Terms

ATN (Augumented Transition Network) - This is one of several parsing strategies used to assign meaning to language input on some computer. An ATN is made up of a set of transition networks with various nodes for representing components of language. For example, a network may have a node for a noun, another for an adjective, a verb or a preposition. The nodes of the network are connected by links or arcs. Conditions associated with the arcs are used to augment the network as some language string is analyzed. For example, one condition might be "assign subject". When a transition is made from one node (a noun) or state to another, and the condition is met, then the ATN can assign the category "subject" to the noun analyzed.

Backtracking - This is a search network used by parsers to solve ambiguity when a sentence may have multiple meanings. If a node in some network is visited but does not meet a specified condition (a failure), this search technique allows the next adjacent node to be visited. Backtracking lets the computer "go back" and search alternative derivations for a sentence.

Bottom-up Parsing - This type of parsing is data-driven. The parser begins by looking at the constituent values (i.e., "boy", "runs") in a sentence (the data) and matches them to their nonterminal symbols (i.e., "noun", "verb"). No predictions are made and the parser must go through an analysis of all constitutents in a string before generating a legal parse. This means that this strategy alone can be inefficient since wrong parses can be generated en-route to the final solution.

Constituent Structure - A sentence is made up of various constituents, nouns, verbs, adjectives. The constituent structure for a particular sentence is represented as a hieranchical tree form of phrasal elements that proceed from the top (sentence) to intermediate level (noun phrase, verb phrase) to lower level (nouns, verbs, adverbs, prepositions, determiners) constituents.

Context-free Grammar - A grammar that consists of a set of rewrite rules each of which has a left-hand side and a right-hand side separated by some symbol, generally - ->. The left-hand side of each rule is a nonterminal symbol of the grammar; the right-hand side is a sequence of nonterminal symbols and terminal symbols. Nonterminal symbols are usually surrounded by angle brackets, for example: < symbol > - -> < one or more symbols >.

Deterministic Parsing - This parsing strategy doesn't "look ahead" or consider other search alternatives while parsing a string. A deterministic parser uses the information available at the time of analysis of some sentence. Since alternative search methods are not used, only one meaning for a sentence is generated.

Expert Model - A representation or knowledge base in an ITS for the domain skill (i.e., foreign language, radar mechanics). Knowledge is represented as rules or schemas, for example. The expert model knowledge is used to teach students in ITS lessons.

Functional Grammar - A list of rules that provide a functional description (subject, object, preposition) for elements in some language string. These rules of functions are made up of a set of attributes and values. Each

attribute value pair represents a function or grammatical relation. For example, an attribute, SUBJECT might have a value, PRONOUN.

<u>Grammar</u> - A set of rules that determines which strings of words are legal sentences in a language. These rules must also specify the syntactic structure for the sentences. Generally, phrase structure rules are used to define a grammar.

<u>Intelligent Tutoring Systems (ITS)</u> - An advanced form of computer-assisted instruction that adapts instruction to the individual. Artificial Intelligence technology is used to represent an expert model of the knowledge domain and a student's developing model. Both models are compared during learning and instruction adapted accordingly so the student's knowledge will approach that of an expert in the field.

<u>Nonterminal Symbol</u> - An element that represents a syntactic word class in a language. For example, noun (N) and verb (V) are nonterminal symbols.

<u>Parsing</u> - Parsing is an algorithmic operation performed by some device (human or machine) on some representation of a natural language (i.e., a sentence). The result is the computation and assignment of structural relations between words after they have been recognized.

<u>Pedagogical Model</u> - This representation is used to present lessons to students in ITS based on error information and student performance. This model represents the knowledge of an expert tutor according to a specified instructional strategy or teaching principles.

Phrase Structure Rules - Each rule is made up of one or more expressions (phrase or single words) that represent the structure of some sentence. The left-hand side of the rule is made up of one or more expressions that define the expressions one the right-hand side. Expressions are placed on the right-hand side of the rule in a particular order. For example,

S - -> NP VP (Sentence "is made up of" Noun Phrase + Verb Phrase)

NP - -> DET N (Noun Phrase "is made up of" determiner + noun)

Semantic Grammar - This grammar uses semantic categories for terminal symbols rather than a rule. An example is presented in the report. These grammars combine syntactic and semantic processing within a single framework, either a context-free grammar or an ATN.

Semantic Network - A data structure for representing factual knowledge. In computer science and psychology, the structure takes the form of a graph in which the nodes represent concepts and the arcs or links between nodes represent relationships among concepts.

Student Model - Strategies used in ITS that compare student performance or errors with a representation of the expert knowledge base or model. Overlay, differential modeling, and model tracing are examples of three student modeling strategies.

Syntax - This is concerned with the structure of the strings of symbols that make up the sentences of a language.

Terminal Symbol - An element that represents an actual word defined in the language. For example, "adjective" and "soldier" are both terminal symbols in English.

Top - down parsing - This type of parsing is goal-driven. This strategy expands a rule from the top node, for example, "S" and matches this symbol with the next lower level symbols in the rule, for example, "NP" "VP". Next these intermediate nodes are matched to the lowest level of values or terminals symbols. A top-down parser must predict what constituents a sentence will contain based on the surface phrase structure rules (i.e., S -> NP VP). This method continues until all elements in a string are assigned a value.